(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 January 2002 (10.01.2002)

PCT

(10) International Publication Number
WO 02/03716 A1

(51) International Patent Classification[7]: H04Q 3/66, H04L 12/56, G06N 3/12

(21) International Application Number: PCT/GB00/03482

(22) International Filing Date: 11 September 2000 (11.09.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
00305549.8 30 June 2000 (30.06.2000) EP

(71) Applicant *(for all designated States except US)*: BRITISH TELECOMMUNICATIONS PUBLIC LIMITED COMPANY [GB/GB]; 81 Newgate Street, London EC1A 7AJ (GB).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: CORNE, David, Wolfe [GB/GB]; 38 Roding Way, Didcot, Oxon OX11 7RQ (GB). KNOWLES, Joshua, Damian [GB/GB]; Flat 1, 56 London Road, Reading RG1 5AS (GB).

(74) Agent: NASH, Roger, William; BT Group Legal Services, Intellectual Property Dept., Holborn Centre, 8th floor, 120 Holborn, London EC1N 2TE (GB).

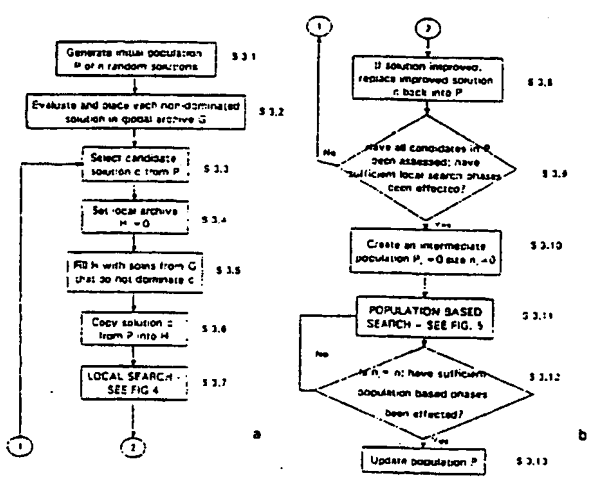
(81) Designated States *(national)*: AU, IN, JP, SG, US.

(84) Designated States *(regional)*: European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published:
— *with international search report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: A MEMETIC METHOD FOR MULTIOBJECTIVE OPTIMISATION



(57) Abstract: A method for multiobjective optimisation comprising the steps of generating a plurality of first solutions; for each of the plurality of first solutions, selecting a first solution and repeatedly modifying the selected first solution so as to generate a second solution; determining optimum configuration parameters represented by one of the first or second solutions for which the cost value is closest to a target value; selecting a plurality of pairs of solutions from the first and second solutions, and for each of the plurality of pairs of solutions repeatedly combining the pair of solutions in accordance with a recombination operator so as to generate a third solution; determining optimum configuration parameters represented by one of the first or third solutions for which the cost value is closest to a target value. Preferably both the modifying step applied to the first solutions, and the combining step applied to pairs of solutions, are repeated a predetermined number of times.

WO 02/03716 A1

A MEMETIC METHOD FOR MULTIOBJECTIVE OPTIMISATION

The present invention relates to optimisation methods and finds particular application in communications network routing.

Many problems involve multiple, often conflicting, measures of performance, which are required to be optimised simultaneously. For example, the problem of routing calls across a network may have at least two objectives: minimise cost and maximise reliability. If each of these objectives were to be optimised independently, as single-objective problems, it is likely that optimal performance for one objective would produce unacceptably low performance in the other. Thus a suitable solution to problems having conflicting objectives should offer acceptable performance in all objective dimensions.

Most real-world multi-objective problems, just as most real-world single-objective problems, are "NP" type problems. This means that no algorithm is known to exist which can guarantee optimal results in a reasonable amount of time; as a result "approximate methods" are required to solve the problems. As most of the "real-world" problems fall within the category of NP-hard problems (a subset of "NP" problems, which indicates that these problems are particularly hard to solve), and thus rely on approximate methods for a solution thereto, this makes the need for good approximate methods for multi-objective problems especially important.

Existing multi-objective optimisation methods are reviewed below, following a glossary section.

Glossary of terms:

Optimisation

Searching through a collection of possible solutions, usually under time constraints, with the aim of finding the best possible solution(s) according to (a) predetermined objective(s) (e.g. cost, reliability etc.).

Single objective problems

A problem in which the measure of a solution's quality is a single objective. For example, if the value of solution A for this objective is 10, and the aim is to minimize, then solution B, which scores 5 for this objective, is a better solution. This

single objective may or may not be a `contrived' objective. That is, it may be a weighted sum or other single-valued measure based upon scores from two or more different objectives.

Multi-objective problems

A problem in which the measure of a solution's quality is a vector of measures relating to two or more objectives.

Solution form

Solutions may be represented by a sequence of values, for example as binary values, in a *chromosome*, each binary value comprising the chromosome being a *gene*, and each gene having a value referred to as an *allele*.

Local search schemes/methods

A small change (*mutation* – see below) is applied to a current solution, so as to move from the current solution to a nearby solution. Different types of local search will differ in terms of the criteria used to determine whether or not the "nearby" solution then becomes the "current solution". This procedure is repeated many times, in search of (a) solution(s) having optimal fitness (see below). Examples of local search methods include simulated annealing and hill climbing.

Population based schemes/methods (Genetic algorithms)

A population of *parent* solutions interact with each other to produce a population of *child* (or *offspring*) solutions: the selection of parent solutions for interaction is often dependent on their respective *fitness* (see below), and the scheme by which they interact (e.g. type of cross-over) is dependent on the problem. In addition to inter-solution interactions, mutations, selected randomly, can be applied to the children. The new population of offspring solutions is then considered as a population of parents for the next iteration of the method. This is repeated many times in search of (a) solution(s) having optimal fitness.

Effecting interaction between parent solutions is described as applying a *recombination operator* to the parent solutions in the following description.

Memetic algorithm schemes/methods

A hybrid of local search and genetic algorithm methods.

Mutation

Take an existing solution and make a small change to it – e.g. for a binary chromosome, only change one or two genes (from 0 to 1, or vice-versa).

Evaluation function

A problem-specific function that evaluates the quality of the solution to the problem.

Fitness

A relative measure of how good a solution is *relative to other solutions.* Fitness is assigned to solutions, and for genetic algorithms (GA), is generally related both to the evaluation of the solution of interest and to the other solutions in the current population: e.g. for single-objective problems, rank solutions in order of quality, then assign fitness in a decreasing order. Fitness can also be used to decide which solutions should be parents: in most GA schemes, the better quality solutions may be given more opportunities to mate and generate child solutions.

Search Space

Range of valid solutions that can be assessed, via the corresponding evaluation function, for solution quality.

Multi-objective scalarising method

When a problem has multiple objectives, it has an evaluation function that comprises contributions from each individual objective. These individual objectives may be combined into a scalar function according to some understanding of the problem – e.g. weighting each of the individual objectives. This therefore enables a multi-objective problem to be expressed as a psuedo-single objective problem, and fitness can be assigned to the scalar as described above. For more information, refer to "An overview of Evolutionary Algorithms in Multiobjective Optimization", Fonseca & Fleming, Evolutionary Computation 3(1) pp. 1 – 16, MIT Press 1995.

Multi-objective Pareto method

The evaluation function corresponding to a multi-objective problem may alternatively be expressed as a vector e.g. $f(x) = (f_1(x) + ... f_n(x))$. In order to compare evaluation functions between solutions, and thus to assign fitness to solutions, the whole vector is compared.

. For example, comparing a pair of solutions, Δ , 4 outcomes are possible:

1. Δ is not worse on any objective than E and is better on at least one

   i.e. $f_1(\Delta) >= f_1(E)$, $f_2(\Delta) >= f_2(E)$, $f_n(\Delta) > f_n(E)$

   In this situation, Δ is said to *dominate* E

2. E *dominates* Δ

3. The solutions or at least the $f_k(i)$ are identical for Δ and E;

4. Δ and E are nondominated i.e. $f_1(\Delta > f_1(E)$, but $f_2(E) > f_2(\Delta)$

This comparison method is known as *"Pareto Dominance"*. In multi-objective optimisation, the overall aim is to find a collection of solutions that are not dominated by any others (including each other). In so-called "Pareto-based" methods for multi-objective optimisation, the Pareto-dominance method is employed in assigning fitnesses to solutions based on relative dominance. Comparing solutions to see which is dominant always involves looking at all of the objectives at once.

For more information refer to Goldberg (1989) Genetic algorithms in search, optimization and machine learning, Reading, MA, Addison-Wesley.

Selection Pressure

Selection of solutions is dependent on the fitness criteria applied to solutions. The pressure of selection is the degree to which solutions with higher fitness values are selected (e.g. fitness threshold). Note that with local search methods, selection of solutions is for replacing old solutions; with population based methods, selection of solutions is for selection of parents AND selection of child solutions to replace old solutions. Thus selection pressure is specified by the fitness required in order to reproduce and/or survive.

Deceptive Landscapes

Assuming a problem to be single-objective, any search strategy basically searches on a mountainous landscape trying to find the highest peak (best quality solution). Considering a landscape such as that shown in Figure 1, A is the highest peak and the best solution. If the search strategy were to start at point C, the search would assess which direction would be best to start moving towards, and would probably search in the direction of B. Even if the strategy were able to see as far as the rightmost base of the 'A' mountain, it is still not as high at that point as it is an equivalent amount to the right of C. The search strategy would repeat this

assessment and movement, each time checking the neighbourhood (by mutating the solution) and moving to the most promising place in the neighbourhood (choosing one mutant to be the new current solution). In this case, a simple search strategy would go all the way to B, and be stuck there.

This landscape is deceptive, in the sense that the information from mutants in the neighbourhood will tend strongly to lead away from the truly good solutions.

Due to the fact that many problems in the real world have multiple objectives, a number of workers have recently been developing methods that can be applied to such problems: e.g. A. J. Chipperfield and P. J. Fleming. **Multiobjective Gas Turbine Engine Controller Design Using Genetic Algorithms**, *IEEE Transactions on Industrial Electronics*, 43(5), October 1996; . S. Chang, etal **Genetic Algorithm Based Bicriterion Optimization for Traction Sustations in DC Railway System**, In *Proceedings of the Second IEEE International Conference on Evolutionary Computation*, pages 11-16, Piscataway, New Jersey, 1995. IEEE Press; Alain Cardon etal **Using Genetic Algorithm in Job-Shop Scheduling Problem to Constraints Negociators' Agents.** In *Proceedings of Evolutionary Algorithms in Engineering and Computer Science, EUROGEN'99*, pages 20-27, Jyväskylä, Finland, May 1999; M. S. Bright and T. Arslan. **Optimal Supply Voltage Selection through a Multiobjective Design Strategy**, In *EEE 33rd Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, California, October 1999. More references may be obtained from URL http://www.jeo.org/emo/EMOObib.html.

The table below shows a *non-exhaustive list* of the types of methods that have hitherto been applied to solve multi-objective problems.

	Local	*Population*	*Memetic (local & population)*
Pareto	Knowles & Corne (1999) The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation, In *1999 Congress on Evolutionary Computation*, pp: 98-105, Washington, D.C., IEEE Service Center	Goldberg (1989) Genetic algorithms in search, optimization and machine learning, Reading, MA, Addison-Wesley Horn and Nafpliotis (1993) Multiobjective Optimization using the Niched Pareto Genetic Algorithm, Technical Report IlliGAl Report 93005, University of Illinois at Urbana-Champaign Srinivas and Deb (1994) Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, *Evolutionary Computation*,2(3):221-248 Zitzler and Thiele (1999) Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach, *IEEE Transactions on Evolutionary Computation*, 3(4): 257-271	
Scalar	Czyzak & Jaszkiewicz (1998) Pareto simulated annealing--a metaheuristic technique for multiple-objective recombination optimization. *Journal of Multi-Criteria Decision Analysis*, 7:34-47; Gandibleux, Mezdaoui, Freville; Pilegaard Hansen (1997) Tabu Search in Multiobjective Optimisation : MOTS. In *Proc MCDM'97*, Cape Town, South Africa	Bentley & Wakefield Finding Acceptable Solutions in the Pareto-Optimal Range using multiobjective GA Schaffer (1985) Multiple objective optimization with vector evaluated genetic algorithms: *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93-100. Lawrence Erlbaum, 1985	Jaszkiewicz (1998) Genetic local search for multiple objective recombination optimization *Technical Report RA-014/98, Institute of Computing Science, Poznan University of Technology* Murata & Ishibuchi (1996) Multi-Objective Genetic Local Search Algorithm, In Toshio Fukuda and Takeshi Furuhashi, editors, *Proceedings 1996 International Conf. on Evolutionary Computation*, pp119-124, Nagoya, JP, IEEE

TABLE 1

The box relating to Pareto and Memetic has no entries, as use of the Pareto method to assign fitness in local search methods has received relatively little attention. In fact, there has been little interest in applying local search methods in any form to multi-objective problems, mainly due to prevailing academic view that *"conventional optimisation techniques, such as gradient based and simplex based methods, and also less conventional ones, such as simulated annealing, are difficult to extend to the true multi-objective case, because they were not designed with multiple solutions in mind....evolutionary algorithms, however, have been recognised to be possibly well suited to multi-objective optimisation since early in their development..."* (Fonseca and Fleming (referenced above)).

However, for some problems, local search methods have been shown to be faster and more efficient than population based methods for single objective problems, as described by Mann and Smith in "A comparison of heuristics for telecommunications traffic routing", published in *Modern Heuristic Search Methods*, Pub Wiley and Son Ltd, and indeed work has been carried out applying scalar methods to local search strategies for multi-objective problems, as shown in Table 1. However, hitherto, there has been little or no development of Pareto methods for local search strategies.

On a practical level, applying Pareto methods to assign fitness is counter-intuitive, as most local search methods store and compare two solutions only – the current and the mutant solution. If two solutions only are assessed for dominance, they are frequently non-dominated with respect to one another i.e. neither is better than the other on all objectives, and it is therefore not possible to judge which of the current and mutant solutions is better.

The scheme presented in "The Pareto Archived Evolution Strategy (PAES): A new baseline algorithm for Pareto multi-objective optimisation" (reference given above) presents a system that attempts to overcome this limitation. PAES stores a plurality of non-dominated solutions, instead of just the current and mutant solution. These solutions are used as an aid to selection between the current and mutated solution because they act as an approximation to a "front" of current non-dominated solutions, thereby providing selection pressure for subsequent solutions. This method thus paves the way for applying Pareto methods to local search schemes, but, and as illustrated in the above-referenced paper, the PAES scheme performs rather poorly in

deceptive problems, suggesting that it will also perform rather poorly in real-world scenarios.

Memetic methods have been shown to be very effective on a number of single-objective problems, and on this basis, several workers have developed memetic methods for solving a range of multi-objective problems (see Table 1). The method developed by Jaszkiewicz, referenced above, aggregates the vector of multiple objectives into a scalar measure (as described above under multi-objective scalarising method). None of the memetic schemes currently available use the Pareto scheme to assess multi-objective evaluation functions. In fact, Jaszkiewicz states that Pareto schemes are unsuitable for multi-objective problem solving due to limitations that he has identified in the scheme itself.

All of the methods discussed above and presented in Table 1 provide approximate means for solving complex problems. Evaluation and comparison of the methods is heavily dependent on the problem on which the method is tested (including the mechanics of the search strategy – choice of parameters etc). When introducing a new method, it is therefore difficult to define any one issue to which such a new scheme is directed. Nevertheless, as new schemes are developed, which, when tested under identical conditions to other approximate methods, perform at least as well, if not better, than existing methods, they provide an alternative tool for solving complex problems, and may potentially solve problems which existing methods are unable to solve. It should thus be clear that as well as presenting the mechanics of a new method, a new method should also be presented in the context of one or more problems on which it has been tested.

According to a first aspect of the invention, there is provided a processor-implemented method of determining optimum parameters of a model of a physical system. The model has a cost value associated with it and one or more operational objectives. The model is formulated in such a way that when the cost value is evaluated, it is evaluated according to each of the operational objectives, such that evaluation of the cost value comprises an evaluation component corresponding to each objective. The method comprises the steps of

objectives, task

generating a plurality of first solutions;

for each of the plurality of first solutions, selecting a first solution and repeatedly

modifying the selected first solution so as to generate a second solution;

determining optimum configuration parameters represented by one of the first or second solutions for which the cost value is closest to a target value;

selecting a plurality of pairs of solutions from the first and second solutions, and for each of the plurality of pairs of solutions repeatedly

combining the pair of solutions in accordance with a recombination operator so as to generate a third solution;

determining optimum configuration parameters represented by one of the first or third solutions for which the cost value is closest to a target value.

Preferably both the modifying step applied to the first solutions, and the combining step applied to pairs of solutions, are repeated a predetermined number of times.

Conveniently, the method may be used for controlling the configuration of a physical system, such as a network transport system comprising a plurality of communication links, associated switches and routing means. When the method is applied to such a system, the optimum configuration parameters determined in the method are used to optimise routing traffic, and the output from the method is input to the routing means.

network

Preferably, the criteria for determining whether a cost value is closest to a target comprises the following steps:

for each of the objectives comprising the model, identifying which of the solutions has a higher evaluated solution component;

identifying which of the solutions has the most number of higher evaluated solution components,

such that if one of the solutions can be identified as having a greater number of higher evaluated solution components, it has a cost value that is closer to the target.

Advantageously the step of determining optimum configuration parameters includes identifying a group of first solutions and comparing the cost value of the

second solution with cost values corresponding to the identified group of first solutions, so as to identify whether to determine said optimum configuration parameters in accordance with said second solution or with said selected first solution.

Further aspects, features and advantages of embodiments of the present invention are described below.

A method of optimising configuration parameters, in particular routing data, in a communications network will now be described, by way of example only as an embodiment of the present invention, and with reference to the accompanying drawings, in which:

Figure 1 is a schematic diagram of an example evaluation function for a single objective problem;

Figure 2 is a schematic diagram of connections between nodes on an SDH network, which, as one of its functions, routes telephone calls between nodes;

Figure 3 is a block diagram of a method of optimising configuration parameters for a system having one or more objectives;

Figure 4 is a block diagram of the local search phase comprising part of the method of Figure 3;

Figure 5 is a block diagram of the archive comparison step comprising part of the local search phase of Figure 4;

Figure 6 is a schematic diagram of results of the evaluation function, plotted in objective space; and

Figure 7 is a block diagram of the population based search phase comprising part of the method of Figure 3.

Overview

objectives

As shown in Figure 3, the method of optimising configuration parameters for systems having one or more operational objectives comprises a local search phase S 3.7 and a population based search phase S 3.11, where the local and population search phases alternately operate on a set of solutions. In the local search phase, an archive of solutions H is maintained. When a solution c is mutated in the local search phase to generate a mutant solution m, each of the solutions within the archive is

compared with that mutant solution, and acceptance or rejection of the mutant solution m is dependent upon comparison of the mutant m with all of the solutions in the archive H. If the mutant solution m is accepted, it replaces the originating solution c in the archive H. The criteria for accepting the mutant solution m may be dominance, such that the mutant solution m is only accepted if it dominates a predetermined number of solutions in the archive.

In one configuration, the mutant solution m may be accepted only if it dominates all of the solutions in the archive H. In this situation, at the end of the local search phase for all of the solutions in the solution set, the archive H comprises the "best seen" solutions.

The local search phase S 3.7 may comprise a plurality of mutation phases for each solution in the solution set, and archive H is preferably freshly populated with solutions at the beginning of each mutation phase. This approach is unlike that of existing systems, where the archive H is populated at the start of each local search phase S 3.7, and mutant solutions are compared with, and added to, the same solutions for each mutation phase. Populating the archive H at the start of each mutation phase allows less constrained searching (as described in detail below), enabling better solutions to be found in areas where there are lots of solutions, and more solutions to be generated in unpopulated areas.

Overview of embodiment problem: Offline routing problem

Figure 2 shows a network G 100, having nodes n 101 and links m 103 therebetween, where n belongs to the set N of nodes, and m belongs to the set E of links. Communication signals are routed through the network 100 by routing data which is usually stored at the nodes. In a traditional voice carrying network, the nodes (or "switches") are provided with routing tables and voice signals are routed at each node towards the next node along a predetermined route to a destination such as the local exchange for a dialled number. The routing tables will include preferred routing data plus secondary routing data if the preferred route is not available due to congestion or failure for instance.

To change the routing, the nodes are reconfigured by updating the routing data at the nodes.

It is known to use shortest path routing between end points in the network. However, other factors can be very important, particularly in more complex communications environments in which both voice and data need to be accommodated. Furthermore, the data requirements can vary from narrow bandwidth to something sufficient to support interactive and visual environments.

Each link 103 has a bandwidth b(m) and a cost c(m) associated therewith, and each of the links 103 are bi-directional. In a traditional telephone network, when a connection is requested, the network attempts to assign a circuit of fixed bandwidth between the source and destination e.g. node 1 and node 5, implemented by means of the routing data. In a data network, the routing for a communications signal is more flexible. The data may for instance be carried in a series of packets, which in practice follow different routes through the network and have to be buffered at the receiving end so that they can be ordered consecutively before being delivered to the user. However, there will still be predefined constraints on routing, for instance to minimise the traffic by selecting shortest paths as far as possible. This also arises in virtual private networks where specified routes are assigned in software to virtual networks for the use of specific customers.

Routing of calls may thus be taken on different timescales, either real-time routing or offline routing. In real-time routing, connections are routed on a moment-by-moment basis, and in offline routing connections are booked in advance. The latter has practical benefits, as a network provider can use predetermined traffic profile information to marry connection requirements with bandwidth availability. As a result, quality of service can be guaranteed and fewer communications fail, such as voice calls having to be turned away (i.e. fewer callers receiving the "engaged" tone).

The offline routing problem to be solved by embodiments of the present invention can be expressed as follows: to route multiple traffic requests r such that:

a) no link is over-capacitated,

b) communications costs associated with use of a link are minimised, and

c) link utilisations are all below a specified, fixed target utilisation,

and therefore addresses 3 objectives of the offline routing problem.

The bandwidth capacities of links in the network are of two types: a backbone type, having a capacity of 64 units (nodes 1 and 4 in Figure 2), and a local

type having, a capacity of 16 units (nodes 2, 3, 5-8). This may conveniently be expressed in the form (using standard set notation):

$\{b(m) \mid m \square E\}$ lie in $\{16, 64\}$

For a set R of r communications which must be routed over network G 100, each communication $r \square R$ specifies a source node $v(r)$ and $w(r)$ (e.g. nodes 1 and 5 in Figure 1). Associated with each communication r, there is also a connection time $T_i(r) \square T$, a disconnection time $T_j(r) \square T$, and a communication bandwidth $h(r)$.

Table 2: Summary of terms

SYMBOL	DESCRIPTION	REPRESENTATION ON FIGURE 2
G	Network	101
N, n	Set of nodes, individual node	101
E, m	Set of links, individual link	103
R, r	Set of communication requests, individual comms request	
u	Target utilisation for link	
X,x	Set of feasible solutions, a solution	
X^*, x^*	Set of Pareto optimal solutions, a Pareto optimal solution	
v(r)	source node	1
w(r)	destination node	5
h(r)	communication bandwidth	
b(m)	link bandwidth	
c(m)	cost associated with link m	
f(m)	total traffic on link m	
t(v, w)	traffic between nodes v and w	
P(v, w)	Path between nodes v and w	105
$T; T_i, T_j$	Set of time frames; time to connect call, time to disconnect call	

For each $v, w \square N$ there is an amount of network traffic, $t(v, w)$, which must be routed from v (node 1) to w (node 5) in the network. This traffic must all be routed on the same path $P(v, w)$ which has to be determined.

Objective a:

Total traffic $f(m) \quad f_{v,w \square N} \{t(v,w) \mid m \square P(v,w)\}$ which must not exceed the bandwidth of that link: for all $m \square E, f(m)\ \delta\ b(m)$, and can be achieved by minimising the deviation between f(m) and b(m): $\min f_{m \square E} \max\{f(m) \quad b(m), 0\}$

Equation 1

Objective b:

For all possible source/destination pairs, cost of routing all traffic t(v, w) on path P(v, w) between v and w: $\min \underset{v,w\square N}{f} \{t(v,w) \upsilon \underset{m\square P(v,w)}{f} c(m)\}$

Equation 2

Objective c:

Minimise deviation from target utilisation, u, for each link in the network: $\min \underset{m\square E}{f} \max\{f(m) \quad \frac{u \upsilon b(m)}{100}, 0\}$

Equation 3

Embodiment of the method of the invention:

Having defined the problem associated with this embodiment of the invention, the embodiment itself can be described. Continuing with the set notation used above, the multi-objective problem can be defined as:

"minimise" $f(x) = (f_1(x),\ldots.f_k(x))$ Equation 4
subject to x□ X

("minimise", the quotation marks indicates that a single solution will typically not be minimal on all objectives)

The three objective functions defined above thus provide individual components $f_1(x)$, $f_2(x)$, $f_3(x)$ (in this case 3 because we have 3 objectives), for which a Pareto optimal set of solutions, X^* □ X, is to be found. Solutions in the Pareto optimal set are known as *efficient* or *admissible* solutions, because very typically there is no solution for objective values (x, y, z) such that all other solutions are either equal or worse on all 3 objectives. Rather, there is a wide collection of diverse solutions, where no one solution dominates another in the set; these diverse solutions comprise the Pareto optimal set.

The embodiment is a memetic method, which uses a local search phase together with a population-based phase, and periodically employs crossover to recombine distinct local optima identified from the local search phase. Two solution archives are used: a first archive G, which comprises the best non-dominated

solutions found throughout both search phases, and a second archive H that is used as a comparison set in each of the local search phases. The second archive H is cleared at the beginning of each mutation of the local search phase, and filled with solutions, which do not dominate the candidate solution c, from archive G. The minimum and maximum number of solutions stored in the archives is problem-specific; for the offline routing problem, the minimum number of solutions is preferably 50, and the maximum is preferably 250.

Each cycle of local and population based search is repeated a predetermined number of times, and the number of repetitions is chosen with knowledge of the problem to which the method is applied. (For any problem, the method should run through at least one local based and one population based phase).

Each solution is expressed as a chromosome, and each gene position within a chromosome corresponds to a path between 2 points in the network (v, w). Such a path is typically predetermined for each pair of source and destination nodes, such that a "lookup table" is created for each pairs of nodes, listing, in ascending order, the cost, bandwidth and link utilisation, between source node v and destination node w. Table 3 shows a typical lookup table for cost only of a call to be routed between nodes 1 and 5, based on the node inter-connectivity of the network shown in Figure 2.

Table 3

Cost	Order	Path between nodes 1&5
4+3+2=9	1	1,7,6,5
3+2+1+4+3=13	2	1,2,8,3,4,5
4+3+4+3=14	3	1,7,6,4,5
3+4+4+3=14	4	1,2,3,4,5
3+2+4+3+2=14	5	1,2,8,7,6,5
4+4+1+4+3=16	6	1,7,8,3,4,5
3+2+1+4+4+2=16	7	1,2,8,3,4,6,5
3+4+4+4+2=17	8	1,2,3,4,6,5
3+4+1+4+3+2=17	9	1,2,3,8,7,6,5
3+2+4+3+4+3=19	10	1,2,8,7,6,4,5
4+4+2+4+4+3=21	11	1,7,8,2,3,4,5
3+4+1+4+3+4+3=22	12	1,2,3,8,7,6,4,5

The embodiment is presented in more detail in Figure 3 of the accompanying drawings, and comprises the following steps (shown for one cycle of local search phase/population based search phase only):

- ξ S 3.1 Create an initial random population P;
- ξ S 3.2 Evaluate all members of the population according to Equation 4, thus evaluating each member on each of the objectives, and place all non-dominated solutions in a global archive G. For the problem presented above, equations 1 – 3 are calculated for each member. The fitness of each member is then always a vector of three elements. The following example illustrates this evaluation process (recall that all of the objectives are considered together):

  Consider 3 candidate solutions, A, B and C, whose fitness vectors are:
  A [100, 100, 100]
  B [50, 80, 90]
  C [40, 110, 70]

  Analysing these solutions for dominance (as defined above), only B dominates A, because it is better than A in every sense. Comparing C/A, C beats A in two objectives but is beaten by A in the third objective, and considering C/B, C beats B on two objectives, only to be beaten by B on the third objective.
- ξ S 3.3 Select a candidate solution c from P;
- ξ S 3.4 Set local archive H=0;
- ξ S 3.5 Fill H with solutions from G that do not dominate c;
- ξ S 3.6 Copy solution c from P into H;
- ξ S 3.7 Apply local search phase to solution c (see below);
- ξ S 3.8 If c is improved as a result of the local search phase, replace c in P with improved c (i.e. higher f(x));
- ξ S 3.9 If there are some candidates in P that have not been passed through the local search phase, go to S 3.3 and repeat steps S 3.3 through S 3.9 until all candidates have been searched.
- ξ S 3.10 Create an intermediate population P_i and set $P_i=0$, size $n_i=0$;

- ξ S 3.11 & 3.12 Apply population based search phase for parents from initial Population P (see below), inputting offspring, or original solutions into P_i until $n_i = n$;
- ξ S 3.13 Update population P, based on P_i

As shown on Figure 3, both the local search phase S 3.7 and the population based search phase S 3.11 are typically repeated many times within this single cycle of local/population based memetic method.

Figure 4 shows a block diagram of the procedure for the local search phase, which maintains a single "current" solution, and, via a form of the hill climbing procedure, searches the space of solutions by continually generating and testing mutants of the current solution. In certain circumstances, a mutant solution becomes the "new" current solution and the search continues from there. An archive, H, is maintained which comprises a representative collection of all of the non-dominated points found thus far by the algorithm. Thus in detail, the method comprises the following steps:

- ξ S 4.1 Mutate current solution c to generate new candidate solution m (apply mutation operator);
- ξ S 4.2 Evaluate m;
- ξ S 4.3 If c dominates m, discard m, and go to S 3.9 else go to S 4.4;
- ξ S 4.4 If m dominates c, replace c with m and add m to archive H; else go to S 4.5;
- ξ S 4.5 Compare m with members of archive H (see below);
- ξ S 4.6 Add m to archive G if it is accepted into archive H.

Figure 5 shows a block diagram of the procedure for evaluating whether m is dominated with respect to members of the archive, and comprises the following steps:

- ξ S 5.1 Assess whether m is dominated by any member of archive H, if so discard m and go to S 3.9, else
- ξ S 5.2 Assess whether m dominates any member of the archive; if no go to S 5.5, else go to S 5.3
- ξ S 5.3 Remove all dominated members of the archive

- ξ S 5.4 Add m to the archive. Go to S 5.10
- ξ S 5.5 Assess whether H is full; if no go to S 5.8, else go to S 5.6
- ξ S 5.6 Assess whether m would increase diversity in H. This is measured by comparing f(x) values. Referring to Figure 6, for each solution in H, the corresponding f(x) values 601a-e are plotted in objective space, within an objective space that has been divided into grid squares 603a. If there are many solutions 601a,b,c within the same grid square 603a, then the solutions within that square are not diverse, relative to one another. However, if a solution 603e falls within a different grid 603b to the grid 603a occupied by the other solutions, this solution 603e is said to be diverse with respect to the other solutions 603a,b,c.

If m is not diverse go to S 5.9, else go to S 5.7

- ξ S 5.7 Replace the member of H residing in the most crowded grid location with m;
- ξ S 5.8 Add m to H;
- ξ S 5.9 Assess whether m is located in a less crowded region of the objective space than the current solution c (see S 5.6) ; if no go to S 5.11 else go to S 5.10
- ξ S 5.10 Accept m; go to S 4.6
- ξ S 5.11 Reject m; go to S 3.8

Steps S 4.1 – S 4.6 (thus including those of Figure 5) are repeated a predetermined number of times so as to give the solution a chance to mutate into an "improved" solution. Thus if candidate solution m is archived into H and G, it becomes the current solution c when steps S 4.1 – S 4.6 are repeated. Typical termination criteria may include number of failures (i.e. number of times the mutant is dominated by the current solution), and/or a fixed number of cycles.

When archive H is populated in step S 3.4, it may be populated by a "representative", rather than a complete, set of non-dominated solutions, due to size limitations (computational limitations). In this case, such a representative set should be selected to provide uniformity of spread of solutions along the current approximation to the non-dominated Pareto surface.

The use of archive H, and the strategy for acceptance of solutions therein, affects the direction of the search for solutions. The procedure described with reference to Figure 5 describes a set of rules that govern solution acceptance. These rules can be expressed by the following generalised decision question:

"My current solution is X, but my new mutant is Y -- shall I go to Y (take that small step on the mountain range) and continue from there, or shall I stay at X (stay put and try out possible steps in other directions)? "

The archive H helps to answer the question. Basically, if Y turns out to be a step into an area of the trade-off space which is already "covered" in the archive H better than X's area, then it may be better to try some new directions from X. However, if going to Y seems to take the search into an area that is relatively under represented by solutions, then it may be better to progress in the direction of Y. It may typically turn out that after a while the archive has lots of points in one area of the trade-off surface but few in others: e.g., between [10, 1] and [6,5] there may be 30 or 40 points on the trade-off surface in the archive ([7.4, 3.8], [9.1, 2.2], etc.), while between [6,5] and [1,10] there may just be a handful of solutions.

The choice of solutions in archive H thus directly affects the search for solutions in the heavily populated area: if, for each round of local search, the archive H remains substantially unchanged (only changing by the replacement of a "better" mutation solution), then according to the selection procedure of Figure 5, lots of mutant solutions will be thrown away until one is found that is a bit closer to the unpopulated area than X is. This strategy promotes the general result of ending up with a good spread of solutions across the entire trade-off surface, but it restricts search in the populated area. However, it may be that allowing the search to progress in the populated area _may_ lead to:

1) even better progress in the populated area -- e.g., the [9.1, 2.2] solution may be found as a mutant of X, and further mutants of this new current solution may have produced something like [9,1], which would remove some solutions in the archive (such as [10,1] and of course [9.1,2.2]), leading to a better result in that trade-off area;
2) even better progress in the unpopulated area - a mutant of (or mutant of a mutant of) the [9.1,2.2] solution may turn out to score [3,6], which could represent strong progress in the unpopulated area - perhaps even better than

what could have been achieved by looking only at mutants of things already in the unpopulated area.

In an attempt to minimise any bias of the search direction and allow unrestricted exploration of solutions, archive H in this embodiment is re-set for each solution local search phase (step S 3.4), and H is populated by solutions that do not dominate the candidate solution (step S 3.5) prior to the search (figure 4). As the initial population of candidate solutions is generated at random, each local search of candidate c is expected to progress relatively freely.

Archive G meanwhile maintains an up-to-date record of the best solutions that have been found at any stage of the search process.

Figure 7 presents the steps comprising the population based search phase, corresponding to S 3.11 on Figure 3. Recall that at S 3.8, a final "improved" solution from the local search phase replaces the current solution in population P (or if the termination criteria has been exceeded without m dominating c, the current solution does not change). Once this has occurred for all of the solutions in P that were randomly generated at step S 3.1, these improved solutions become parents for the population based search phase. Referring to Figure 7, the population based search phase comprises the following steps:

- ξ S 7.1 Randomly chose 2 parents within archive G
- ξ S 7.2 Combine parents to form offspring c. The parents may be combined according to one of many established population based methods – e.g. crossover between genes of the parent chromosomes.
- ξ S 7.3 Compare c with solutions in archive G;
- ξ S 7.4 If c is dominated by solutions in archive G, discard c and use binary tournament to select new c from G. Binary tournament is a known competitive method of selecting a solution from a population of solutions: typically, two solutions are chosen at random from archive G and their f(x) values are compared; the solution with the "minimised" f(x) is then selected to be the offspring c (note that this is different to most tournaments, where solutions are typically selected to be parents for subsequent reproduction to produce offspring);

If c is not dominated by solutions in G go to S 7.5

ξ S 7.5 Assess whether c is in a more crowded grid location than parents (refer to S 5.6); if it is not go to S 7.6, if it is, go to S 7.1

ξ S 7.6 Place offspring c into intermediate population P_i

Thus a child resulting from S 7.2 is accepted only if it is nondominated with respect to the entire archive G and if it resides in a less crowded region than at least one of its parents. If it dominates any member of archive G then it is also accepted. Solutions that are dominated by member(s) of archive G, or that reside in more crowded regions are rejected. In this case two new parents are selected (S 7.5) and combination is applied again (S 7.1). The procedure is repeated until either a child is accepted, or a predetermined threshold number of combinations is exceeded.

From the afore-described conditions of selection, it can be seen that this embodiment of the invention is elitist.

Modifications

The local search phase that is described with reference to Figures 4 and 5 could be modified such that steps S 4.1 through S 4.6 could be revised to implement local search techniques more akin to, for example, tabu search or simulated annealing. These steps, in their unmodified form, most resemble the simple local search scheme called "hillclimbing", in that a mutated solution is only retained as the new "current" solution if it is clearly better than the old current solution in some reasonable sense.

In the embodiment described above, one of the conditions for accepting a modified solution is that the modified solution is better than the current solution. As a result, solutions from the local search phase are primarily *exploited* – that is to say that the embodiment concentrates on exploiting solutions that have already been used to generate nearby solutions. This strategy, which maintains a high selection pressure, performs little *exploring* of the search space, which, for deceptive problems, can be crucial to finding the optimal solution (referring back to Figure 2 – peak A is unlikely to be found from local exploitation of peak B). One way of reducing selection pressure is to relax the condition of storing best solutions, or of strict elitism, as a condition for solution acceptance. For example, in the local search

phase, the method could be modified such that new solutions are compared to a *subset* of archive H rather than all of archive H. As an alternative, a simulated annealing style acceptance function could be used, whereby solutions selected at random that are worse, rather than better, may be accepted. Thus in alternative local search schemes (simulated annealing being a good example), worse mutated solutions could sometimes be accepted as the new current solution; or the mutation operation (S 4.1) might be restricted to ensure that the mutated solution is suitably different from other mutations of the current solution which have been applied recently.

The population-based phase provides an opportunity to explore the search space more, because it uses more information to generate offspring. However, the elitist selection criterion maintains high selection pressure during this phase as well as during the local search phase. In the population based search phase, removing step S 7.4 – i.e. the step of rejecting solutions produced by crossover, could reduce elitism. Furthermore, the type of combination between parents could deliberately be selected to reduce selection pressure.

The procedure described above with reference to Figure 5, for establishing whether a solution is in a "crowded" grid location (described in S 5.6), could be replaced with the known fitness sharing approach, whereby the distance between solutions is used to evaluate the relative degree of intra-solution crowding.

In the embodiment described above, parents in the population based search phase are combined via recombination (S 7.1). This combining operation is problem-specific, and may be replaced by any one of the following alternatives (non-exhaustive list): apply crossover as described above, and then mutate one or more genes of the resulting offspring; code the parents as real-valued parameters and combine them by taking the average of each allele from the two parents for every gene; apply multi-parent crossover; apply differential evolution; apply particle swarm; apply optima linking; or apply population based incremental learning. These are standard techniques, and more information can be found in "New Ideas in Optimisation", D. Corne etal, McGraw Hill.

Furthermore, the random selection of parents at S 7.1 may alternatively be replaced by selecting two parents via binary tournament solution, and then considering their "reduced surrogate" (which is the collection of loci at which the

parents have different allele values). Tournament selection can then be used among genes in the reduced surrogates, and a locus can be chosen whose allele in at least one of the parents is relatively high. A child can then be formed by replacing the lower valued allele at this locus with the higher one, with that child otherwise being the same as the parent that contained the lower-valued allele. This is illustrated below in Table 4:

Table 4

Parent 1	1	2	1	1	1	3	1	5	1	1	1	2	1	4
Parent 2	1	2	4	1	2	3	1	1	1	1	9	2	1	7
Reduced surrogate			4		2			1			9			7
Tournament selection					2						9			7
Winner											9			7
Break Ties											9			
Offspring	1	2	1	1	1	3	1	5	1	1	9	2	1	4

The embodiment described above is concerned with a problem having multiple objectives; however, the invention does not exclude application of the invention to single-objective problems. The method of the invention could be applied to a single-objective problem where the problem has been broken up into units, each of which can be evaluated separately, e.g. Travelling salesperson problem. The objective of this problem is to find a permutation of (say) 10 cities, and given such a permutation (a candidate solution) the single-objective of interest is simply the total distance travelled when visiting those cities one by one in the order given in the permutation. This can be expressed as a two objective problem according to the following two objectives:

f_1: the distance travelled visiting the first five cities in the permutation

f_2: the distance travelled visiting the second five cities in the permutation

Where the single objective of minimising total distance travelled $= w_1 f_1 + w_2 f_2$

Equation 5

The problem can be treated as a multi-objective problem, having objectives f_1 and f_2, and applied to the method described above with reference to Figures 3 – 7. Once the Pareto optimal solutions for f_1 and f_2 have been found, f_1 and f_2 can be recombined according to the single objective weighted function, Equation 5, yielding a range of optimum single-objective values. These resulting solutions are likely to include the optimal solution to the single-objective problem.

Very complex single-objective problems may lend themselves to such analysis, as this approach breaks up a problem into several more manageable units. This may increase exploration of the search space (searching the space of solutions in different ways), whilst still guaranteeing as good a solution as is possible.

To demonstrate further aspects of the method of the invention, a second problem is briefly presented below.

The problem is the Adaptive Distributed Database Management Problem (ADDMP), as disclosed in the Applicant's co-pending application, International publication number WO00/08569, published 15th February 2000 (Applicant's case ref A25664), and the reader is referred to the afore-referenced publication for full details of the problem. Briefly, distributed data service providers (DDSP) offer a database service to a collection of clients, both of which are often globally distributed. There is therefore a network of nodes (client, server, both), where each node may generally provide an entry point to a LAN, which contains a server provided by the DDSP. Essentially, one of the functions of the DDSP is to ensure that database users (clients) receive an adequate quality of service (QoS) when accessing data from a database (server).

There is a great range of client/server arrangements that fall within the ADDMP: numbers of clients and servers can range typically between 2 and 20, and the number of servers between 10 and several thousand. Database access patterns can very equally dramatically, depending on the type of data to be retrieved (e.g. financial markets), and the time of day. The ADDMP is thus the problem of finding the best client/server connection configuration, given a particular scenario, which specifies details of the underlying communications network, server speeds, and access rates for each client. Published application WO00/08569 presents a single-objective optimisation strategy, whereas the current embodiment is concerned with a

multi-objective strategy. Two objectives for this problem may be to minimise worse delay and median delay in client/server access time.

For the ADDMP, each gene in a chromosome represents a server that a client machine should use. For example, for chromosome

3, 1, 2, 6, 2, 8, 4, 4, 10, 1

the first client is to use server 3; the 2nd client is to use server 1; the 3rd client is to use server 2; the 4th client is to use server 6 etc. i.e. each position along the chromosome 'belongs' to a specific client, and its allele (i.e. the value at that position) tells the 'owning' client which server to use.

For the ADDMP, the local search phase is identical to that described above with reference to Figures 4 and 5. The population-based phase, described with reference to Figure 7, differs slightly, in that S 7.1 preferably comprises standard uniform crossover together with a mutation function. The mutation function picks a gene at random in the offspring, and changes it to a random new value.

The off-line routing problem and the ADDMP are two examples of a class of problems that may be generalized in terms of their components and the function of those components. In general, these problems have a plurality of nodes and communication channels between at least some of the nodes, and data is transferred from a source to a destination via various communication channels in accordance with routing logic that is in communication with the nodes.

Implementation details:

Apparatus to effect the method of the above embodiment may be loaded on a terminal running the Unix operating system (e.g. Sun workstation running Solaris), and the method may be embodied in a computer program written in the C programming language. The choice of the C programming language is inessential to the invention and any low-level programming language could be used; furthermore the method could be run on any operating system on any single-processor hardware. Alternatively, the method could be parallelised.

When the invention is used for routing in a network, the output from the method may be a plurality of suggested routes between nodes v_i and w_i. The terminal running the optimising method may be connected to nodes on the network to enable

the output from the optimisation to be cascaded to as many of the nodes as necessary, for use as routing data. In order to download this output to the nodes, the terminal may also include means for communicating with the nodes, means for presenting the suggested routing information in a format that can be received by control programs running on the nodes and means for sending the information between the terminal and the nodes.

As will be understood by those skilled in the art, the invention described above may be embodied in one or more computer programs. These programmes can be contained on various transmission and/or storage mediums such as a floppy disc, CD-ROM, or magnetic tape so that the programmes can be loaded onto one or more general purpose computers or could be downloaded over a computer network using a suitable transmission medium.

Unless the context clearly requires otherwise, throughout the description and the claims, the words "comprise", "comprising" and the like are to be construed in an inclusive as opposed to an exclusive or exhaustive sense; that is to say, in the sense of "including, but not limited to".

CLAIMS

1. A processor implemented method of determining optimum parameters of a model of a physical system, the model having a cost value associated therewith, which model has one or more operational objectives and is formulated in such a way that when the cost value is evaluated, it is evaluated according to each of the operational objectives, such that evaluation of the cost value comprises an evaluation component corresponding to each objective, the method comprising the steps of
   i. generating a plurality of first solutions;
   ii. for each of the plurality of first solutions, selecting a first solution and repeatedly
      a) modifying the selected first solution so as to generate a second solution;
      b) determining optimum configuration parameters represented by one of the first or second solutions for which the cost value is closest to a target value;
   iii. selecting a plurality of pairs of solutions from the first and second solutions, and for each of the plurality of pairs of solutions repeatedly
      a) combining the pair of solutions in accordance with a recombination operator so as to generate a third solution;
      b) determining optimum configuration parameters represented by one of the first or third solutions for which the cost value is closest to a target value;
   iv. repeating steps ii and iii a predetermined number of times; and
   v. outputting the optimum configuration parameters

2. A method according to claim 1, including controlling the configuration of a physical system in accordance with the optimum configuration parameters.

3. A method according to claim 2, wherein the physical system is a network transport system comprising a plurality of communication links, associated switches

and routing means, in which the method further comprises loading the optimum configuration parameters to the routing means.

4. A method according to any one of the preceding claims, in which the criteria for determining whether a cost value is closest to a target comprises the following steps:

a) for each of the objectives comprising the model, identifying which of the solutions has a higher evaluated solution component;
b) identifying which of the solutions has the most number of higher evaluated solution components,

such that if one of the solutions can be identified as having a greater number of higher evaluated solution components, it has a cost value that is closer to the target.

5. A method according to any one of the preceding claims, in which the determining step (step ii (b)) includes identifying a group of first solutions and comparing the cost value of the second solution with cost values corresponding to the identified group of first solutions, so as to identify whether to determine said optimum configuration parameters in accordance with said second solution or with said selected first solution.

6. A method according to claim 5, in which the identified group of first solutions is stored in a first store.

7. A method according to claim 6, in which a second solution for which the cost value is closest to a target replaces its corresponding first solution in the first store.

8. A method according to any one of the preceding claims, in which the identified group of first solutions further includes the first solution from which the second solution derives.

9. A method according to any one of the preceding claims, in which the plurality of first solutions is randomly generated.

10. A method according to any one of the preceding claims, in which step ii (a) of modifying a first solution comprises applying a single change to the first solution.

11. A method according to any one of the preceding claims, in which the recombination operator used in step iii (a) on a pair of solutions comprises any one, or a combination of, crossover between the solutions, and/or mutation.

12. A method according to any one of the preceding claims, in which the plurality of first solutions is stored in a second store.

13. A method according to claim 12, in which the second and third solutions for which cost values are closest to the target replace their corresponding first solutions in the second store.

14. A processor implemented method of determining optimum parameters of a physical system, the method comprising the steps of

i. generating a plurality of first solutions representing configuration parameters of the physical system;

ii. modelling the physical system using the configuration parameters to determine a cost value associated therewith, which system has one or more operational objectives and is formulated in such a way that when the cost value is evaluated, it is evaluated according to each of the operational objectives, such that evaluation of the cost value comprises an evaluation component corresponding to each objective,

iii. for each of the plurality of first solutions, selecting a first solution and repeatedly

   c) modifying the selected first solution so as to generate a second solution;

   d) determining optimum configuration parameters represented by one of the first or second solutions for which the cost value is closest to a target value;

iv. selecting a plurality of pairs of solutions from the first and second solutions, and for each of the plurality of pairs of solutions repeatedly

c) combining the pair of solutions in accordance with a recombination operator so as to generate a third solution;

d) determining optimum configuration parameters represented by one of the first or third solutions for which the cost value is closest to a target value;

v. repeating steps ii and iii a predetermined number of times; and

vi. outputting the optimum configuration parameters.

15. A method according to claim 14, including controlling the configuration of the physical system in accordance with the optimum configuration parameters.

16. A computer program, or a suite of computer programs, comprising a set of instructions to cause a computer, or a suite of computers, to perform the method according to any one of claims 1 to 13.

17. A computer program, or a suite of computer programs, comprising a set of instructions to cause a computer, or a suite of computers, to perform the method according to claim 14.

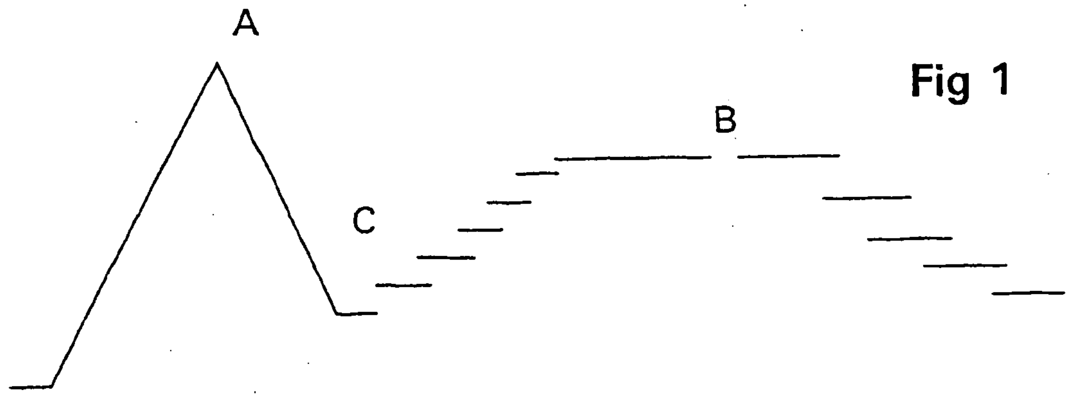

A
B
C

Fig 1

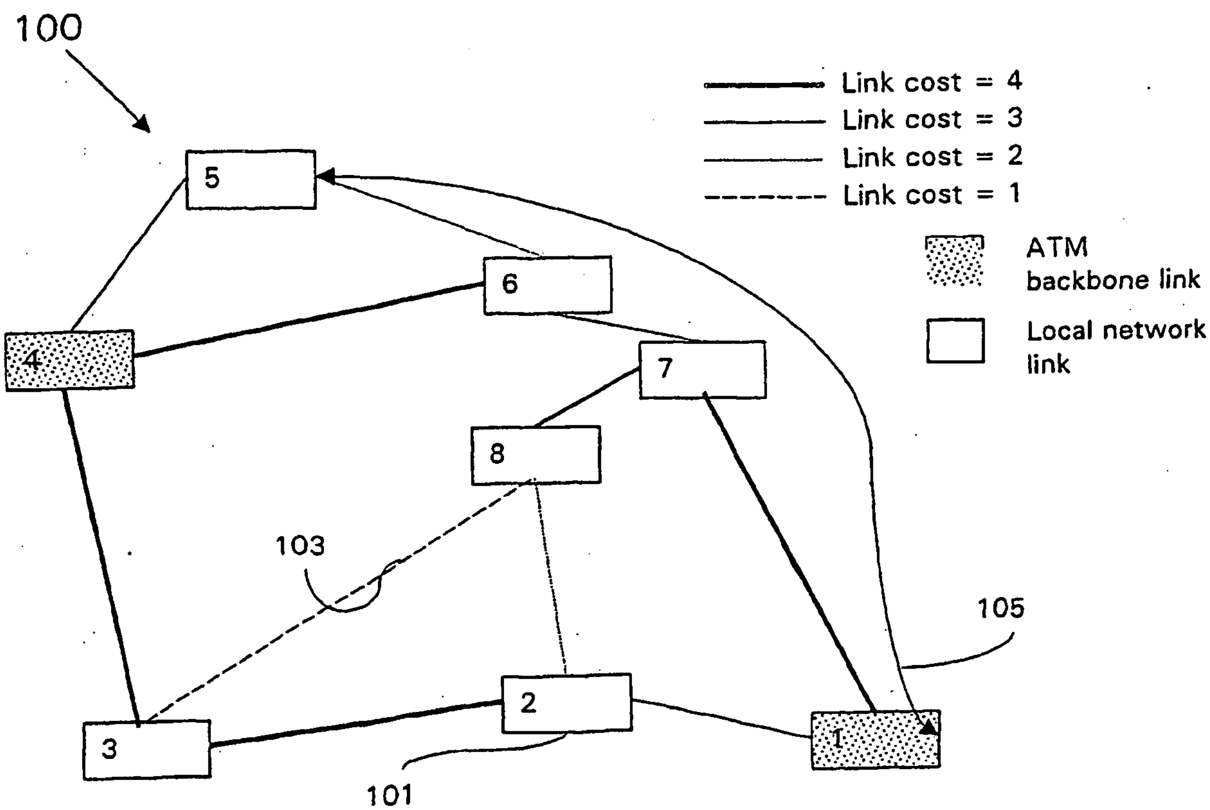
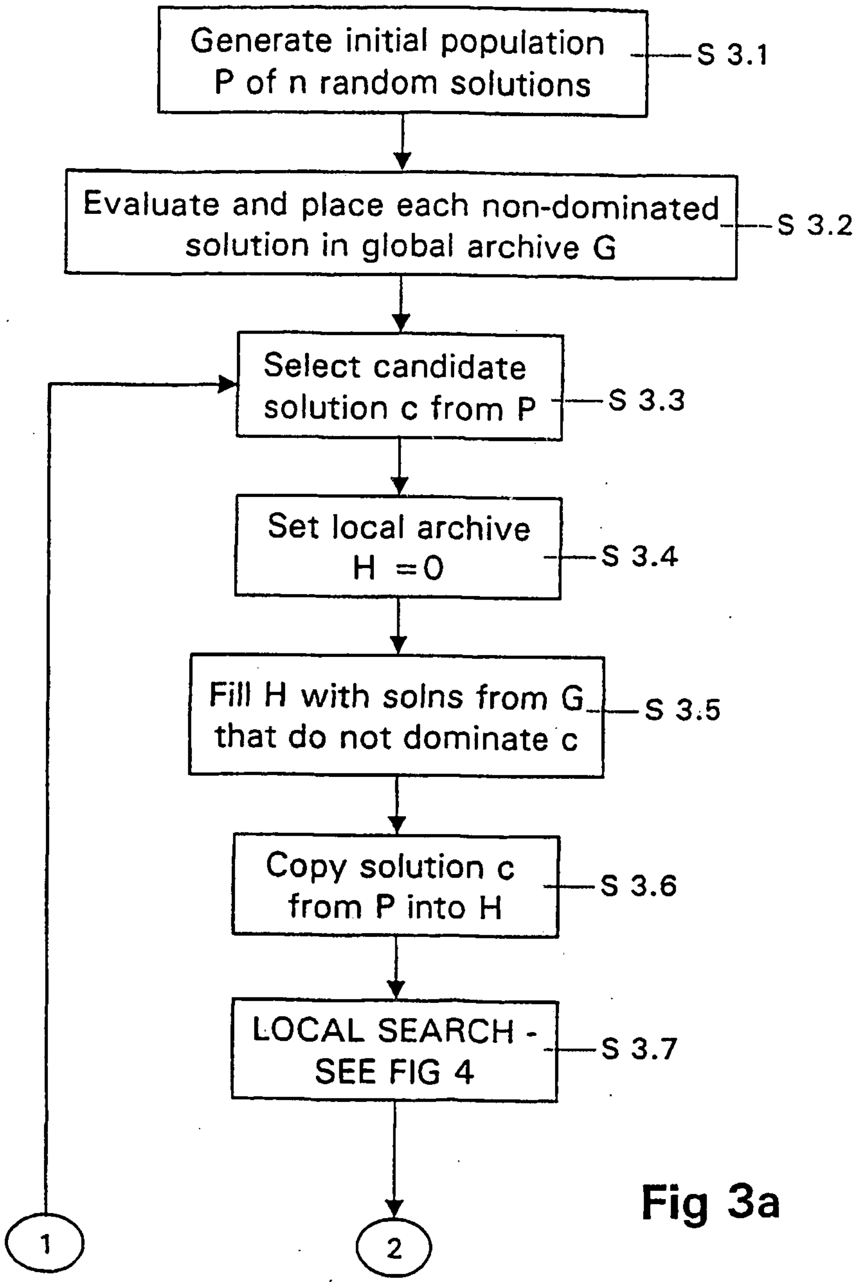

100
Link cost = 4
Link cost = 3
Link cost = 2
Link cost = 1
ATM backbone link
Local network link
5
6
4
7
8
103
105
3
2
1
101

Fig 2



Generate initial population P of n random solutions
S 3.1
Evaluate and place each non-dominated solution in global archive G
S 3.2
Select candidate solution c from P
S 3.3
Set local archive H =0
S 3.4
Fill H with solns from G that do not dominate c
S 3.5
Copy solution c from P into H
S 3.6
LOCAL SEARCH - SEE FIG 4
S 3.7
1
2

Fig 3a

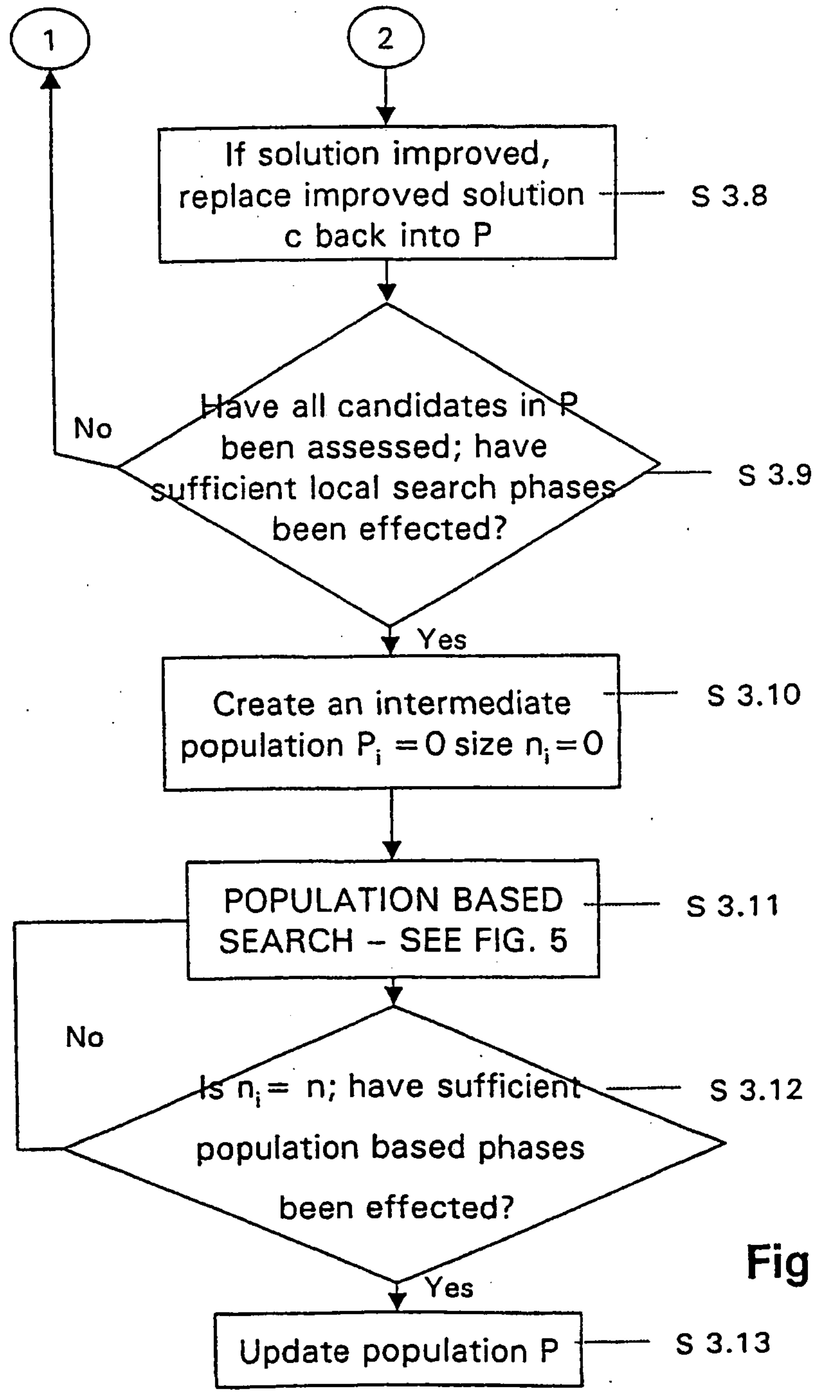
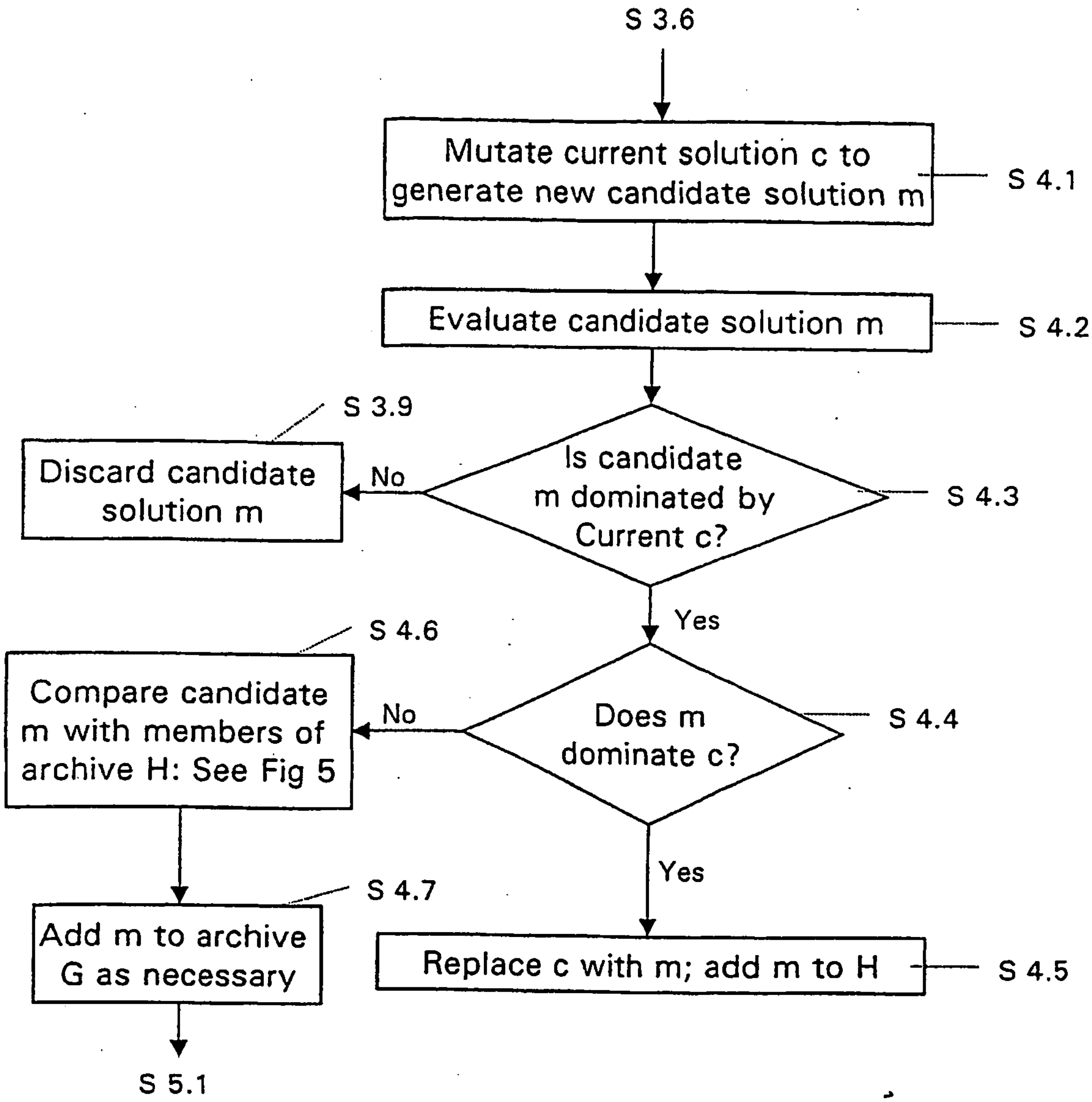

1
2
If solution improved, replace improved solution c back into P
S 3.8
Have all candidates in P been assessed; have sufficient local search phases been effected?
No
Yes
S 3.9
Create an intermediate population $P_i = 0$ size $n_i = 0$
S 3.10
POPULATION BASED SEARCH – SEE FIG. 5
S 3.11
Is $n_i = n$; have sufficient population based phases been effected?
No
Yes
S 3.12
Update population P
S 3.13

Fig 3b



S 3.6
Mutate current solution c to generate new candidate solution m
S 4.1
Evaluate candidate solution m
S 4.2
S 3.9
Discard candidate solution m
No
Is candidate m dominated by Current c?
S 4.3
Yes
S 4.6
Compare candidate m with members of archive H: See Fig 5
No
Does m dominate c?
S 4.4
Yes
S 4.7
Add m to archive G as necessary
Replace c with m; add m to H
S 4.5
S 5.1

Fig 4

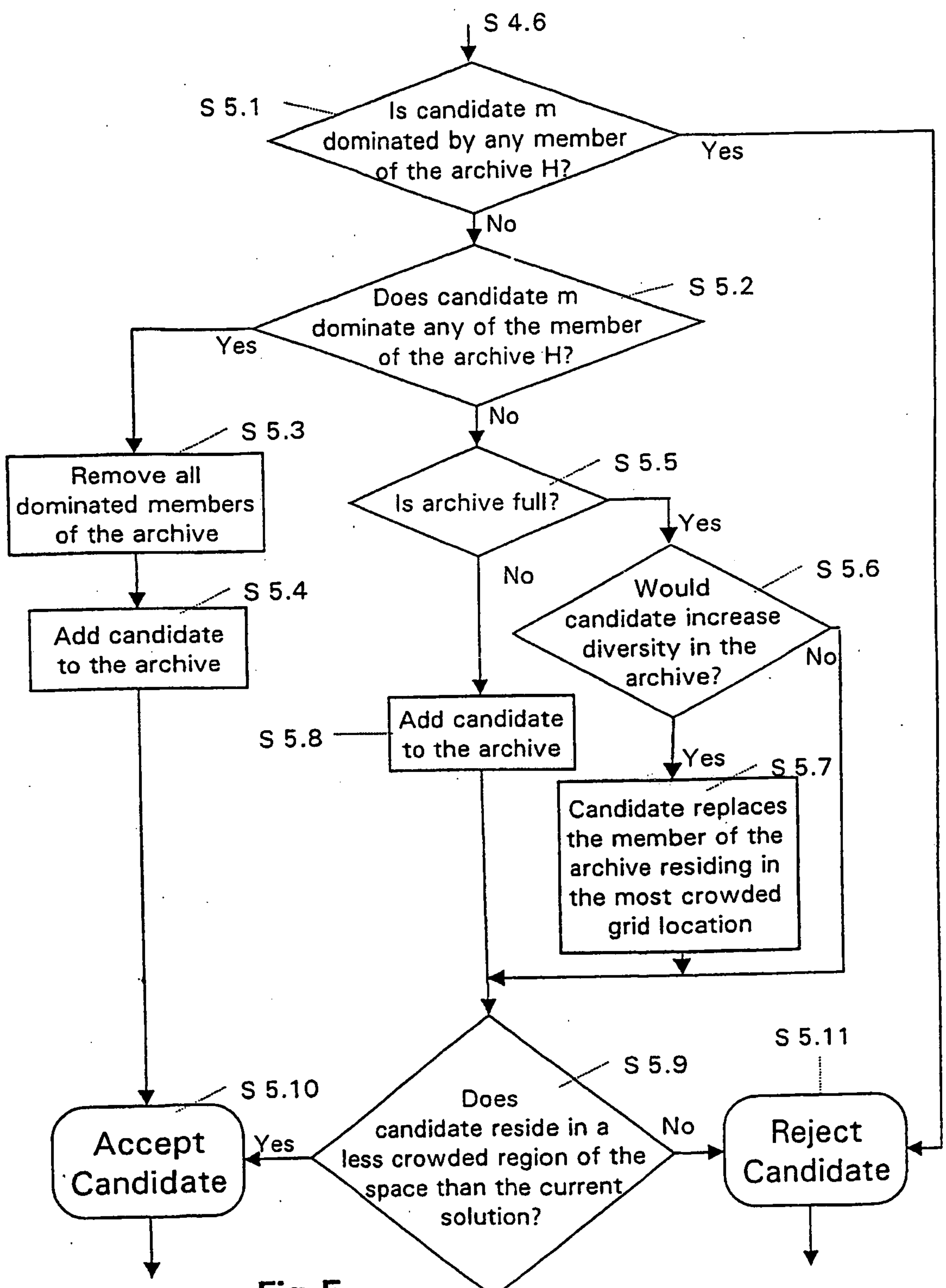

S 4.6
S 5.1
Is candidate m dominated by any member of the archive H?
Yes
No
S 5.2
Does candidate m dominate any of the member of the archive H?
Yes
No
S 5.3
Remove all dominated members of the archive
S 5.4
Add candidate to the archive
S 5.5
Is archive full?
Yes
No
S 5.6
Would candidate increase diversity in the archive?
No
Yes
S 5.7
Candidate replaces the member of the archive residing in the most crowded grid location
S 5.8
Add candidate to the archive
S 5.9
Does candidate reside in a less crowded region of the space than the current solution?
Yes
No
S 5.10
Accept Candidate
S 5.11
Reject Candidate

Fig 5

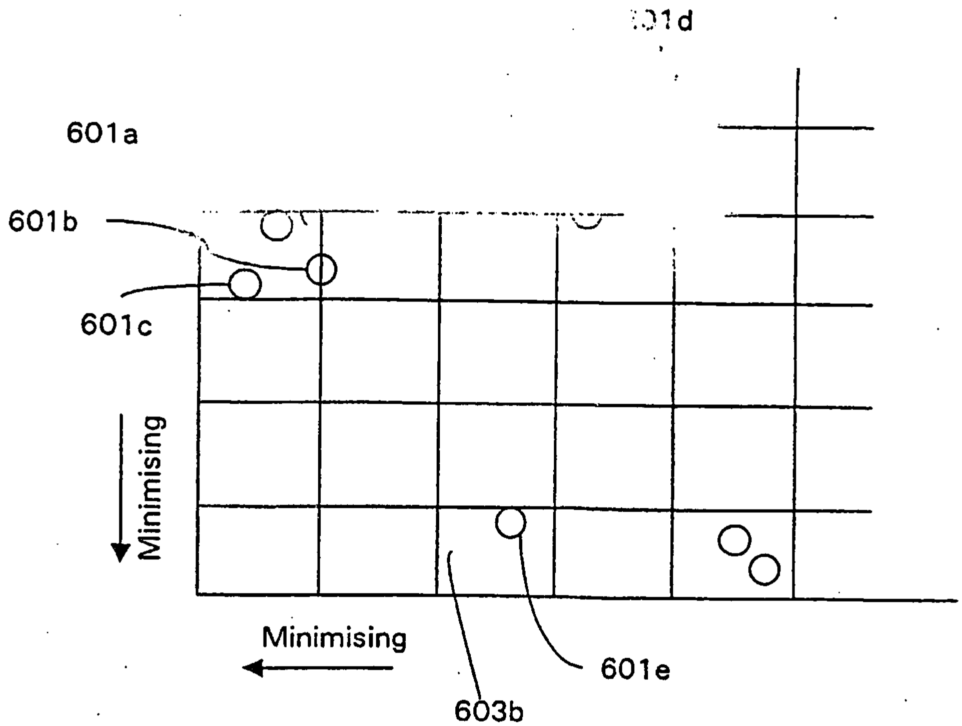


Fig 6

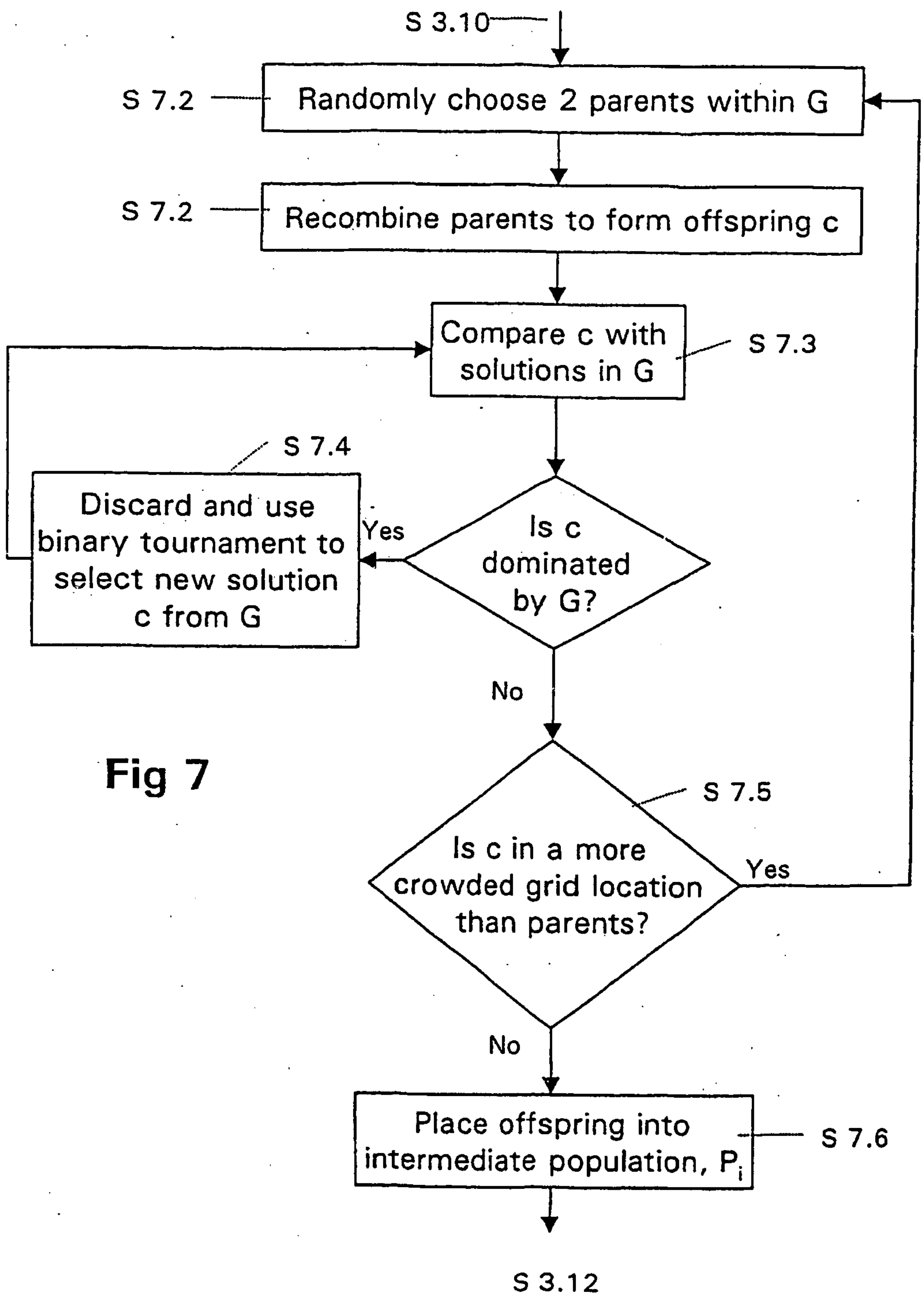

S 3.10
S 7.2
Randomly choose 2 parents within G
S 7.2
Recombine parents to form offspring c
Compare c with solutions in G
S 7.3
S 7.4
Discard and use binary tournament to select new solution c from G
Yes
Is c dominated by G?
No
S 7.5
Is c in a more crowded grid location than parents?
Yes
No
Place offspring into intermediate population, P_i
S 7.6
S 3.12

Fig 7

INTERNATIONAL SEARCH REPORT

Intel   nal Application No
PCT/GB 00/03482

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7   H04Q3/66   H04L12/56   G06N3/12

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7   H04Q   H04L   G06N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, IBM-TDB, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	MERZ P ET AL: "A Comparison of Memetic Algorithms, Tabu Search, and Ant Colonies for the Quadratic Assignment Problem" PROCEEDINGS OF THE 1999 CONGRESS ON EVOLUTIONARY COMPUTATION (CEC'99), WASHINGTON, DC, USA, 6-9 JULY 1999, vol. 1, pages 2063-2070, XP002154390 figure 1	1-11, 14-17
A	---	12,13

-/--

[X] Further documents are listed in the continuation of box C.

[ ] Patent family members are listed in annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
4 December 2000	13/12/2000

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Vercauteren, S

INTERNATIONAL SEARCH REPORT

International Application No
PCT/GB 00/03482

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	KNOWLES J ET AL: "The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Pareto Multiobjective Optimisation" PROCEEDINGS OF THE 1999 CONGRESS ON EVOLUTIONARY COMPUTATION (CEC'99), WASHINGTON, DC, USA, 6-9 JULY 1999, vol. 1, pages 98-105, XP002154391 cited in the application	1-11, 14-17
A	figures 1,2	12,13
A	ISHIBUCHI H ET AL: "Local Search Procedures in a Multi-Objective Genetic Local Search Algorithm for Scheduling Problems" 1999 IEEE CONFERENCE ON SYSTEMS, MAN AND CYBERNETICS (IEEE SMC'99), TOKYO, JAPAN, 12-15 OCTOBER 1999, vol. 1, 12 - 15 October 1999, pages 665-670, XP002154392 * page 666, section 2.1 * * page 667, section 2.5 * page 668, right-hand column, last paragraph	1,14
P,X	KNOWLES J D ET AL: "M-PAES: A Memetic Algorithm for Multiobjective Optimisation" PROCEEDINGS OF THE 2000 CONGRESS ON EVOLUTIONARY COMPUTATION (CEC'00), LA JOLLA, CA, USA, 16-19 JULY 2000, vol. 1, pages 325-332, XP002154422 the whole document	1-17